

Southwest University of Science and Technology

本科毕业设计（论文）

题目名称：基于STM32的智能家居系统设计

|  |  |
| --- | --- |
| 学院名称 | 信息工程学院 |
| 专业名称 | 自动化1701班 |
| 学生姓名 | 林世涛 |
| 学号 | 5120170317 |
| 指导教师 | 于春梅 教授 |

二〇二一年六月

基于STM32的智能家居系统设计

摘要：智能家居技术是在电子信息技术和无线通信技术还有软件技术等技术领域不断发展而产生的新兴技术，这种技术不仅能改变人们日常的生活方式，还能将居住环境变得更舒适。本文的研究设计正是通过上述技术提出一种以无线通信技术为基础实现对室内家居环境进行实时监测、预警、控制的智能家居系统方案。该方案以WiFi技术为核心，使用性价比较高的STM32F1芯片，对装置上的一氧化碳(简称)气体传感器、甲烷()气体传感器 、温湿度传感器、光照传感器人体感应模块进行定期数据采集处理，并发送到系统屏幕和云平台上，云平台扮演数据传输中的数据库和服务器环境。使用云服务器搭建的WEB站点通过向云平台发送指令或接收数据，最终实现了数据在硬件装置上与WEB站点上以云平台为中间件的连接通路，实现智能家居系统的远程查看和智能化控制。

关键词：智能家居；无线通信；环境监测；web技术；传感器；处理器；

**Design of intelligent home system based on STM32**

**Abstract:** Smart home technology is an emerging technology that has been developed in the fields of electronic information technology, wireless communication technology and software technology, which can not only change people's daily life style but also make the living environment more comfortable. The research design of this paper is to propose a smart home system solution based on wireless communication technology to realize real-time monitoring, warning and control of indoor home environment through the above technology. The program uses WiFi technology as the core, and uses the cost-effective STM32F1 chip to regularly collect and process data from the carbon monoxide (CO) gas sensor, methane (CH\_4) gas sensor, temperature and humidity sensor, and light sensor human sensor module on the device, and send it to the system screen and cloud platform, which plays the role of database and server in data transmission environment. The WEB site built using the cloud server sends instructions or receives data to the cloud platform, and finally realizes the connection path between the data on the hardware device and the WEB site with the cloud platform as the middleware to realize remote viewing and intelligent control of the smart home system.

**Key words:** Intelligent Home System; Wireless Communication; Web-technology; sensor; processor;

目 录

[第1章 绪 论 1](#_Toc72532827)

[1.1 概述 1](#_Toc72532828)

[1.2 国内外研究现状 1](#_Toc72532829)

[1.3 研究目标和内容 2](#_Toc72532830)

[第2章 功能设计和方案选择 4](#_Toc72532831)

[2.1 整体方案设计分析 4](#_Toc72532832)

[2.2 无线数据传输方案选型 4](#_Toc72532833)

[2.2.1 无线通信技术 4](#_Toc72532834)

[2.2.3 服务器、数据库实现方式 5](#_Toc72532835)

[2.3 硬件方案选型 5](#_Toc72532836)

[2.3.1 硬件总体设计 5](#_Toc72532837)

[2.3.2 控制器选型 6](#_Toc72532838)

[2.3.3 硬件显示屏选型 6](#_Toc72532839)

[2.3.4 Wifi模块选型 7](#_Toc72532840)

[2.3.5 烟雾传感器选型 7](#_Toc72532841)

[2.3.6 温湿度传感器选型 8](#_Toc72532842)

[2.3.7 光照传感器选型 8](#_Toc72532843)

[2.3.8 人体感应模块HC-SR501 8](#_Toc72532844)

[2.4 本章小结 9](#_Toc72532845)

[第3章 硬件电路设计 10](#_Toc72532846)

[3.1 硬件结构设计 10](#_Toc72532847)

[3.2 硬件原理图设计 11](#_Toc72532848)

[3.2.1 电源电路 11](#_Toc72532849)

[3.2.2 TFT模块显示电路 11](#_Toc72532850)

[3.2.3 ESP8266-01S应用电路 12](#_Toc72532851)

[3.2.4 STM32F103C8T6核心板电路 12](#_Toc72532852)

[3.2.5 蜂鸣器报警电路 13](#_Toc72532853)

[3.2.6 RGB控制电路 13](#_Toc72532854)

[3.2.7 步进电机控制电路 13](#_Toc72532855)

[3.2.8 DHT22传感器电路 14](#_Toc72532856)

[3.2.9 MQ4传感器电路 14](#_Toc72532857)

[3.2.10 BH1750传感器电路 15](#_Toc72532858)

[3.3 本章小结 16](#_Toc72532859)

[第4章 软件设计 17](#_Toc72532860)

[4.1 软件设计分析 17](#_Toc72532861)

[4.1.1 硬件装置应用程序结构分析 17](#_Toc72532862)

[4.1.2 网站终端展示程序结构分析 18](#_Toc72532863)

[4.2 硬件应用程序设计 18](#_Toc72532864)

[4.2.1 初始化程序设计 18](#_Toc72532865)

[4.2.2 OneNET云平台简介 20](#_Toc72532866)

[4.2.3 MQTT协议简介 21](#_Toc72532867)

[4.2.4 MQTT协议数据包格式 22](#_Toc72532868)

[4.2.5 云平台通信程序设计 24](#_Toc72532869)

[4.2.6 云平台上传与接收程序设计 24](#_Toc72532870)

[4.3 网站程序设计 24](#_Toc72532871)

[4.3.1 搭建服务器环境 24](#_Toc72532872)

[4.3.2 VUE框架简介 25](#_Toc72532873)

[4.3.3 VUE-CLI创建VUE工程 25](#_Toc72532874)

[4.3.4 网站布局设计 26](#_Toc72532875)

[4.4 本章总结 27](#_Toc72532876)

[第5章 结果分析 28](#_Toc72532877)

[5.1 设计指标测试 28](#_Toc72532878)

[5.2 调试过程中遇到的问题 28](#_Toc72532879)

[第6章 总结与展望 30](#_Toc72532880)

[6.1 本文总结 30](#_Toc72532881)

[6.2 工作展望 30](#_Toc72532882)

[致谢 32](#_Toc72532883)

[参考文献 33](#_Toc72532884)

# 第1章 绪 论

## 概述

随着近年来科学技术的快速发展，进一步加快了社会的信息化深入发展，信息化赋能正在向人们生活的每个角落涌去，科技无时无刻不在以一种崭新的面貌出现在人们的视野中，试图改变人们的观念。无论房屋大小，现在人们在居住时都会考虑更多舒适性、安全性，人们希望在严寒的冬天不用起身下床就能关掉头顶的灯，希望厨房的天然气和浴室的热水器不会出现有害气体超标的情况，希望自己不在家时能实时了解室内情况，所以拥有远程控制以及智能化两大特点[1]的智能家居作为一种能直接改善人们居家生活质量的技术一经出现就备受人们的关注。现今不断发展、创新的互联网技术带动了有线、无线通信技术还有软件技术、大数据等相关技术学科向相互融合的方向发展[2]。这些学科的相互融合与人们的不断实践也使得智能家居从单薄的概念逐渐走向丰满和清晰。智能家居的兴起，无疑会带给人们更加全面、细致且舒适、安全的居家体验。

目前，作为一门新兴的科学技术，智能家居技术及其行业发展都还处于初级阶段，提出的很多方案都只是一种尝试性地前行，尚存在很多缺点和不足，如很多都缺乏实质性的产品和内容、采用的技术较为复杂且成本较高、行业内没有指定一个标准和规范，各家厂商都是使用的自己的规范和标准，普通消费者会出现难以选择和比较的窘境。

## 国内外研究现状

20世纪的八十年代，智能家居的概念是在欧美国家最先提出，并且一提出就引起了各国电器公司、相关学者们的关注和讨论 [9]。并且在20世纪八十年代，美国就出现了世界上最早的智能房屋设计[3]，其他公司也都看准智能家居前景，纷纷开始了对于智能家居技术的研究，美国、加拿大、德国、日本等发达国家都开始大力推行智能家居概念。

每个国家由于其国情都不相同，所以智能家居落地的解决方案往往也都大相径庭。比如美国、加拿大，地广人稀，人们基本上是一户一栋，即独立式别墅，所以当地的智能家居公司所提供的解决方案会更多考虑整个房子的布局，且由于枪支私有化，所以安防系统会是整个智能家居系统的重点。再例如1998年5月在新加坡电子消费展览会上公布的具有新加坡特点的智能家居解决方案—“未来之家”[4][5]。它的功能有：智能报警、留言储存、家电控制以及核心信息展示面板等等，功能十分强大。

国外智能家居发展至今，因为基本理念深入人心，各公司的安装售后流程完备、成体系，总体行业发展比较成熟，且各个公司产品细分程度很高，用户完全可以通过自己的需求选择，例如作为第一家上市的智能家居相关公司，Alarm公司最初就是以之前提到的安防需求作为切入点，结合移动互联网、云计算等新兴技术打造了首个家庭无线安防系统，根据2016年Business Insider的调查发现，在他们的样本数据中，有41%的人认为智能家居最大的作用就是安防。而大名鼎鼎的Crestron公司就是比较典型的智能家居公司代表，拥有先进的控制技术和一整套完善的集成方案与产品，涉及了室外遮阳、室内照明、影音、安防、IT、建筑管理等，与超过一半以上的世界500强企业都有合作，可以说是智能家居行业的标杆和典范。

国内的智能家居发展比较晚且非常缓慢，当智能家居概念在国外大行其道时，国内还是如日方升之时，老百姓的经济基础和思想概念都远远达不到让智能家居盛行的条件。随着经济的稳步增长，和人们的需求趋向多样化，智能家居才慢慢走近人们的视线，时至今日，国内的智能家居发展正逐步向国外跟进，国内的设计方案思路大体分为两种类型：有线和无线。有线基本上就是从房子装修阶段参与房屋内部的设计，搭建整体方案[8]，例如深圳的欧瑞博，产品上通过人工智能与物联网技术实现了多模态分布式的主动智能，方案设计上与国外公司相似通过设计线路将居住空间的灯光照明、暖通环境、门锁安防、门窗遮阳等设备连接成完整的智能化系统；无线则多是以轻度的体验为主，通过多种智能单品经过简单的配网就能实现一定程度的智能互联，比如小米的系列产品，种类丰富、齐全，并且通过一个多协议网关就能支持不同通信协议的产品接入，便于管理。

## 研究目标和内容

本设计以无线通信技术为基石，通过各种传感器检测，将输出信号送给主控芯片处理，通过主控芯片将数据传给通信模块，再由通信模块传到服务器，而终端从服务器上实时获取数据并显示，并且终端产生的控制数据也原路返回送入主控芯片做出实时修改控制。

通过实现以上流程完成一个能够实现温湿度检测、烟雾报警、远程控制家电、入户监测功能的简单的智能家居模拟系统设计。设计的具体要求如下：

1. 综合考虑各种因素，进行方案选择和论证；
2. 根据设计目标和指标要求选择传感器、驱动装置等外围设备，设计外围电路；
3. 选择现场与手机的通信方式，进行软件设计，完成设计目标要求的功能；
4. 软硬件联合调试。

具体的指标要求：

1. 温湿度精度1度，2%
2. 烟雾检测浓度超过25 %报警

具体研究的内容如下：

1. 综合考虑各种因素，进行整体框架方案的选择和论证；

2. 通过分析设计目标、指标要求和各传感器厂商提供的数据手册选择传感器、芯片、显示器装置等外围设备；

3. 在无终端情况下硬件装置对环境的测量、显示以及复杂任务的实现；

4. 终端软件和数据库(服务器)的方案选择：开放数据服务管理平台如yeelink、中移物联(onenet)或者自行构建数据服务器与终端。WEB端、手机APP、微信小程序等多终端的深入比较，解决前后端联通；

5. 有关物联网技术和无线连接技术的深入研究(热门方案如ZigBee、WIFI、Z-wave)，选择硬件装置与手机的通信方式，进行装置上软件设计，完成设计目标要求的功能；

6. 软硬件联合调试，对系统进行调试优化，打通软硬件之间的联通，实现检测和控制的最终效果。

# 第2章 功能设计和方案选择

## 整体方案设计分析

智能家居系统需要实现环境监测、烟雾报警、人员入户检测、远程控制家电以及远程查看实时信息等功能。其中，环境监测、烟雾报警和人员入户检测很显然都需要一个核心控制器循环处理上述事件，其次要实现远程控制家电和实时查看数据流信息需要选择一种无线通信技术为基础搭建一个数据远程传输框架。所以下文将会以无线数据传输方案选型和硬件方案选型两方面展开选择比较。

## 无线数据传输方案选型

### 无线通信技术

常见的无线通信技术有：无线局域网（Wi-Fi）、近场通信（NFC）、蓝牙（Bluetooth）、红外线数据通信IrDa和ZigBee。

1. WIFI通信技术：

WiFi是一种基于IEEE802.11系列协议标准实现的无线通信技术，该通信协议于1996年由澳洲的研究机构CSIRO提出，WiFi 凭借其独特的技术优势，被公认为是目前最为主流的WLAN技术标准。随着WiFi无线通信技术的不断优化和发展，当前主要有4种通信协议标准，即802.11g、802.11b、802.11n和802.11a[13]，并且新的协议版本IEEE802.11n完全兼容之前的a、b、g版本[7]。根据不同的协议标准主要有两个工作频段，分别为2.4GHz和5.0GHz。

WiFi具有网络拓扑结构简单、通信安全、工作频段开放、与以太网的兼容性较好、传输速率高等优点，常常被应用于无线热点、手机、远程控制、网络媒体、医疗器械、现代农业等众多领域[14]。

1. 蓝牙通信技术：

蓝牙是一种是一种使用UHF无线电波传输的近距离无线通信技术标准，其传输距离通常在10厘米-10米不等，主要工作在2.4GHz频段，并且其距离越远传输效率越低，越不稳定，蓝牙的最大数据传输速率是1Mbps，其功耗介于wifi和ZigBee之间。是典型的点对点、短距离无线通信方式[15]。

1. ZigBee通信技术：

ZigBee是一种基于IEEEE802.15.4标准的低功耗、短距离的无线传输技术，主要工作于2.4GHz频段，传输速率为10M-250kb/s[6],ZigBee设备之间进行数据通信时是通过将数据一个节点一个节点的传递完成的，同时由于其优越的低功耗和组网过程具备自组织、自维护等特点，多用于工业控制，医疗监护，建筑设计、个人电脑外设的无线连接等领域[10]。

剩下的NFC和红外数据通信技术明显不适用于当前多数据、远距离的设计要求，就不具体展开分析。蓝牙技术在日常生活中通常只是用于设备与设备之间点对点的数据传输，且传输距离较短，并不是很适合。ZigBee技术能进行多个电子设备间组网，且功耗低，传输可靠，应用场景很适合，但是私组网并不能直接接入互联网，而是还要通过WiFi才能接入，并且本设计并没有组网需求，所以ZigBee也并不适合。

WiFi技术尽管也是短距离的无线通信技术，但是当接入宽带的无线路由器设置了AP模式之后，无线路由器的WiFi信号就能连入互联网。

接入互联网之后，将数据上传到服务器，就能实现在网页、APP甚至是小程序上的数据流查看和指令发送。

### 服务器、数据库实现方式

个人搭建具有完备功能的服务器和稳定的数据库，并实现数据互通，API自由调用，对于非专业开发者而言，会十分艰难，所以在查阅了网上资料和相关书籍以及咨询了老师之后，有两种较为简单的方式实现应用服务器功能：

(1).在linux环境下搭建boa web服务器

(2).选择专业物联网云平台接入智能家居设备，利用云平台的公共API获取网页所需数据以及下发指令

尽管boa微服务器的搭建不是很难，但是后续步骤如上传数据流、访问请求和下发指令的相关资料并没有找到，并且OneNET平台功能十分完善，支持多协议接入，对每一个注册用户都有开发的公共API接口供调用，官方提供的SDk还支持多种编程语言。

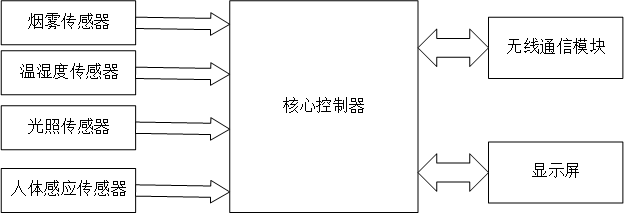
综上所述，最后决定使用开放云平台OneNET作为数据上传的服务器和数据库。

## 硬件方案选型

### 硬件总体设计

根据设计要求和设计指标，在智能家居系统的硬件装置上需要相应指标的传感器、负责收集和处理传感器数据的核心控制器、与外界交换信息的通信模块。并且为了后期调试的方便，应该在硬件装置上加上一个用于实时显示信息能与上传数据作对比的屏幕。

硬件总体结构图如图2-1所示：



**图2-1 硬件结构设计图**

### 控制器选型

根据使用经验，有MSP430、AT89C51、STC15、STM32以及Exynos4412开发板可供选择。由于本次设计所用的传感器数量多，不仅需要给传感器分配引脚，还需要给通信模块、显示屏以及串口调试和SWD下载预留足够引脚，AT89C51引脚太少，难以使用。传感器一般所需供电电压为5v，而MSP430的工作电压范围是：1.8V-3.6V，压差过大，即使做了电压转换也会有IO口电流过大的危险，况且MSP430的I/O无保护，过压过流会立即击穿，所以MSP430也不太适合。Exynos4412使用需要配置linux环境，且绘制PCB比较麻烦，cortex-a9内核功能强大会有过剩。而STC15与STM32相比，后者编写程序使用库函数开发方便，社区文档资料丰富，所以综合以上，最后将控制器定为STM32F103系列。

### 硬件显示屏选型

硬件装置上的显示屏用于前期调试，和后期联通云平台后与云平台的上传数据作对比，因此需要在屏幕上显示尽量多的传感器输出数据和其他的一些实时运行状态。

对于传统单片机开发，有几种常用的屏幕模块可供使用：OLED液晶显示屏、TFT液晶显示屏以及LCD1602显示屏。

根据本小节的显示屏需求分析，OLED屏因为只有0.91或0.96寸等几种有限大小的规格，并不能满足本设计需要尽可能多的显示数据的信息。LCD1602则是一种传统的点阵式液晶显示，1602的正常使用一共需要16个引脚，这无论对于任何单片机而言都是一笔不小的开销，所以并不适合。综上所述，最后显示屏模块选用1.44寸的TFT彩色液晶屏。

### Wifi模块选型

Wifi模块选用国产安信可的ESP8266-01S模块，此模块在较小尺寸的封装中集成了一个32位MCU，可以接入RTOS系统，板载天线并支持标准的IEEE802.11 系列协议，内置 TCP/IP 协议栈[16]。开发者使用AT指令就能轻松使用，接入互联网。与MCU通过串口相连接，如果有过WiFi模块的使用经验，此模块是非常方便的。并且供电电压是3.0-3.6V与主控芯片配合能方便设计电路。

### 烟雾传感器选型

(1).离子式烟雾传感器

离子感烟探测器主要是利用飘散在空气中的烟雾颗粒改变传感器内部的电离室电流原理而设计的探测器。传感器内部的传感器元件装有а放射源的电离室,现在大多为单源双室结构(补偿室,测量室),再配上相应的放大电路和MCU(Microcontroller Unit)所构成。在电离室的正极上有a放射源AM241,其放射源可以在上百年的时间里不断地放射出a粒子，a粒子不断地撞击空气分子，引起电离,产生大量带正,负电荷的离子,从而使空气带有导电性[17]。当传感器内部电路的电源电压加在加在正负极两边时，就会产生电流[11]，在正常情况下电流值会保持稳定。火灾发生时，烟雾粒子进入电离室引起电离电流的减小，电路中通过检测电流或者负载电阻的电压变化就能得到烟雾浓度变化。

(2).光电式烟雾传感器

这种传感器内部通常会含有一个光学迷宫，在光学迷宫两旁安装有红外对管，无烟时红外接收管收不到红外发射管发出的红外光，当烟尘进入光学迷宫时，通过折射、反射，接收管接收到红外光，智能报警电路判断是否超过阈值，如果超过发出警报。

(3).气敏式烟雾传感器

在一般情况下，检测特定种类气体气通常会选用气敏传感器。它主要包括半导体气敏传感器、接触燃烧式气敏传感器和电化学气敏传感器等[12]，其中用的最多的是半导体气敏传感器，其内部使用的是二氧化锡半导体材料，当该材料与烟雾接触时，会引起材料表面电导率的变化，根据传感器厂商给的曲线图资料就能拟合出烟雾浓度与材料等效电阻的关系。它的应用主要有：一氧化碳浓度的检测、煤气中甲烷浓度的检测、乙醇气体浓度的检测等等。

根据设计指标，本设计必须检测出具体的烟雾浓度，但是如果不是针对特定气体的浓度检测，这项功能就失去了意义，而第一种和第二种都不是针对特定种类气体的传感器，并且检测出烟雾浓度后也不能作出具体的区分。综上所述，选用第三种烟雾传感器，并设定此应用场景大多在室内的智能家居系统应当检测的气体为厨房燃火可能产生的甲烷(CH4)、燃烧不充分可能产生的一氧化碳(CO)，传感器的具体型号为——甲烷(CH4)-MQ4、一氧化碳(CO)-MQ7。

### 温湿度传感器选型

温湿度传感器选用的是DHT22模块，这是一个可以同时检测温度和湿度的传感器模块，内部含有一个电容式感湿元件和一个NTC测温元件[18]，并与一个8位MCU连接，拥有高响应、抗干扰能力强等特点。每个DHT22都在极为精确的湿度校验室中进行校准，其温度精度在±0.5°以内，湿度精度在±2%以内。传感器本身采用单总线通信，根据高低电平不同的持续时间制定了通信协议。通常应用于除湿器、数据记录器、家电、气象站等。

### 光照传感器选型

光照传感器选用的是BH1750传感器，检测光照强度通常我们会想到使用光敏二极管，并且光敏二极管的电压/电流随光照强度的增加而相应增加，但是这二者并不是线性关系，而BH1750内部将光敏二极管的输出做了线性处理，传感器芯片与外部(如单片机)采用I²C协议通讯，单片机只要与传感器芯片使用正确格式通信就能得到内部寄存器存储的光照强度数据。

### 人体感应模块HC-SR501

这是一种应用热释电红外传感器LH1778为核心的自动控制模块。

此模块在探测半径范围内有人员经过时，会探测到并输出高电平，在光强稳定无变化时，模块会在输出高电平一段时间之后输出低电平，表示探测范围内无人员。同时模块内置光敏控制和温度补偿机制，前者可根据具体场景需要在白天或光线强时不感应，最大程度降低功耗；当天气升温至30°及以上时会对模块内部传感器有温度影响，后者温度补偿机制设置之后可做一定程度上的性能补偿。

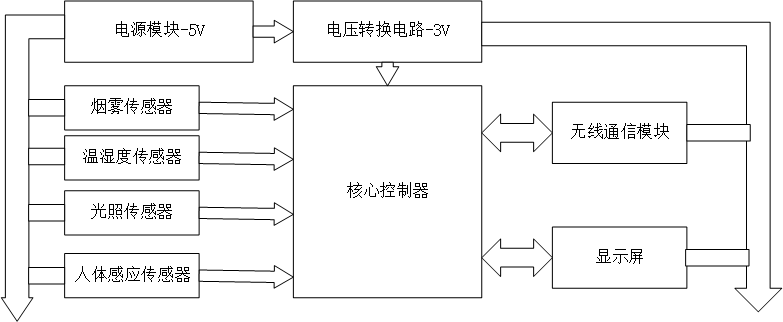
## 本章小结

本章主要是对老师提出的设计指标和具体的设计要求做了详细的分析，并根据以往的单片机开发经验和各元器件特性作了比较，最终选定了无线数据通信的整体方案和硬件设计框架，一一确定了模块的具体型号，为接下来的硬件电路设计踏出了关键的一步。

# 第3章 硬件电路设计

## 硬件结构设计

在设计一个硬件装置电路时，首先应该考虑的是模块和元器件的供电，也就是敲定电源。查询资料并根据上一章选定的控制器和各模块的工作特性，可以得知装置主要分为5V供电和3V供电两个部分，结合实际考虑我们通常使用USB接口接电脑、笔记本或者充电宝为硬件电路供电，所以需要一个设计一个电压转换电路将5V转为3.3V为主控芯片和WIFI模块供电。并且主控芯片作为电路核心部分，参与对各传感器输出数据的搜集和处理并实时传输给显示屏和WIFI模块，控制WIFI模块根据相应协议发送数据。综上，硬件结构设计图如图3-1所示：



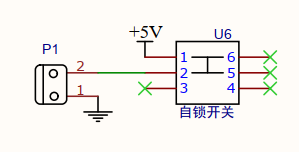
**图3-1 硬件结构设计图**

接下来将一一对各部分电路作具体分析与设计。

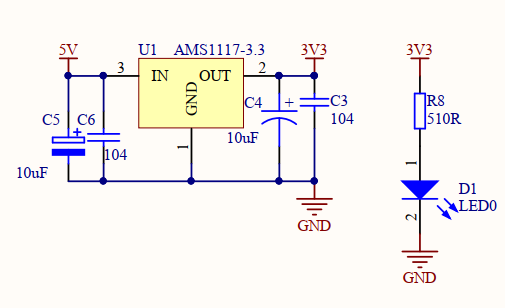
## 硬件原理图设计

### 电源电路

根据前一节所述，装置上使用USB供电除了要引入5V电压配合自锁开关形成基本的电源回路，还必须要有一个电压转换电路才能，主控和其他芯片才能正常工作。



**图3-2 5V电源电路图**

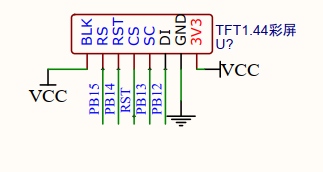


**图3-3 电压转换电路图**

如上图3-3所示，此处使用的是AMS1117芯片作为降稳压芯片并根据芯片手册提供的样例电路设计出了相应的外围电路。AMS1117是一个线性稳压芯片，输出波纹小，工作噪声低,拥有固定输出版本和可调输出版本，这里选用的是固定版本，输入端连接5V能正常输出3.3V。

### TFT模块显示电路

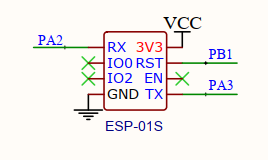
TFT显示屏本身就是一个厂家设计好的显示模块，所以要应用到装置中只需要根据厂家所给芯片手册配置好相关引脚即可。电路图如图3-4所示：



**图3-4 TFT模块显示电路图**

### ESP8266-01S应用电路

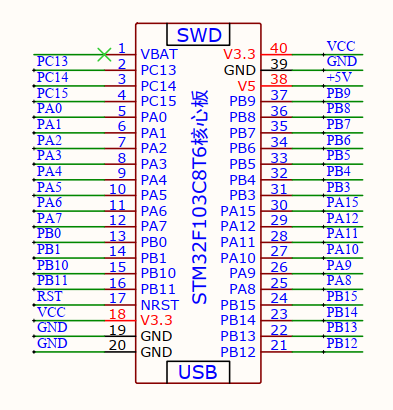
ESP8266与TFT显示屏类似，也是使用厂家设计好的模块，配置相关引脚即可。应用电路图如图3-5所示：



**图3-5 ESP8266应用电路图**

### STM32F103C8T6核心板电路

STM32负责传感器的数据收集和处理，还有ESP8266和TFT两个模块的控制，所以，设计时考虑将STM32的所有IO引脚、电源引脚以及一些常用的功能引脚分别按距离远近引出到两旁的排针上，以免后期如果要增加新的功能或要求时可以灵活变动。核心板电路图如图3-6所示

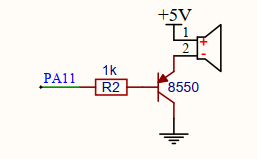


**图3-6 STM32核心电路图**

### 蜂鸣器报警电路

此电路是基于设计要求中的烟雾报警功能而设计的。蜂鸣器采用的是有源蜂鸣器，有源蜂鸣器的有源，指的是蜂鸣器内部设有振荡源，只要能通电，振荡源就能配合电压产生方波信号，有源蜂鸣器就能直接发出声音，无源蜂鸣器则相反[19]。

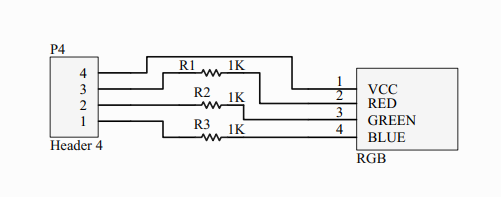
此处设计电路直接使用一个PNP型三极管8550，使用的是三极管的截止区和饱和区实现开关作用。当IO口输出高电平时，三极管导通进入饱和区，当IO口输出低电平时三极管断开进入截止区[20]。报警电路如图3-6所示：



**图3-7 蜂鸣器报警电路图**

### RGB控制电路

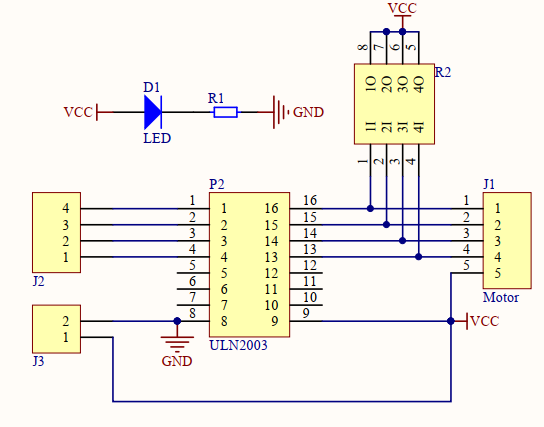
RGB信号灯用于展示智能家居系统的远程控制家电效果——模拟灯光控制。此RGB灯是共阳极，所以只要接通电源，使用IO口控制相应颜色的通道就能展现不同的颜色。RGB控制电路如图3-7所示：



**图3-8 RGB电路图**

### 步进电机控制电路

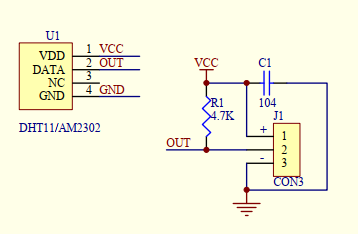
步进电机用于展示智能家居系统的远程控制家电效果——模拟窗帘控制。电路中使用ULN2003集成芯片，输出端能输出大电流，非常适用于驱动电机这类功率要求高的器件。步进电机控制电路如图3-9所示：



**图3-9 步进电机电路图**

### DHT22传感器电路

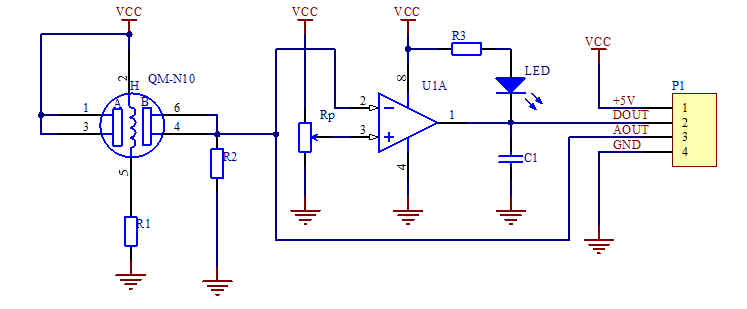
DHT22采用AM2302作为核心芯片，AM2302内部已经集成了电路，所以能直接产生数字输出，根据AM2302芯片手册中的样例电路设计了外围电路。DHT22电路如图3-10所示：



**图3-10 DHT22电路图**

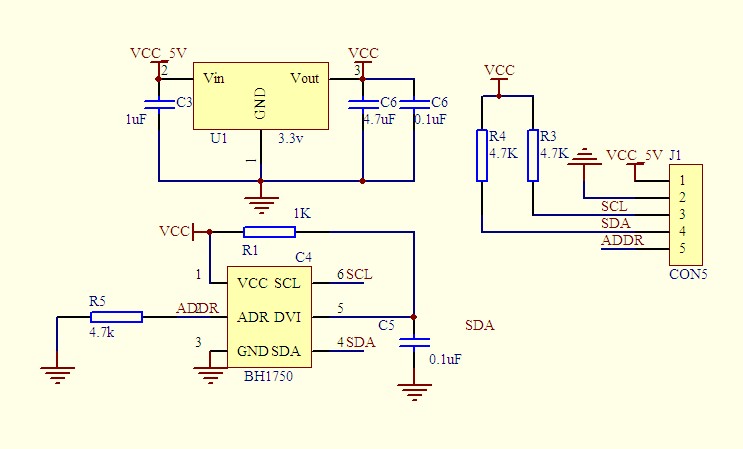
### MQ4传感器电路

气敏式烟雾传感器中最常用的是MQ2传感器，但是MQ2对于每种气体烟雾的灵敏度都差不多，不能在有电平信号时区分到底是哪种气体，所以比较MQ系列的传感器资料得出MQ4传感器对于甲烷的灵敏度较高，可用来检测甲烷气体浓度。MQ4传感器电路设计了数字信号和模拟信号输出，因为需要测量具体浓度信息，所以需要用主控芯片的内置ADC通道采集模拟信号输出。MQ4传感器电路如图3-11所示：



**图3-11 MQ4电路图**

### BH1750传感器电路

BH1750本身已经是设计好的传感器芯片，所以只需要根据芯片手册设计对应的外围电路即可，BH1750传感器电路如图3-12所示：

**图3-12 BH1750电路图**

## 本章小结

本章在第二章的分析设计基础上分部分实现了各模块的电路设计与原理图绘制，主要包括电源电路，主控核心板电路，ESP8266WiFi模块电路，TFT显示屏电路，用于远程控制的RGB灯控制电路和步进电机控制电路以及各种传感器电路。

完成原理图的绘制之后，使用AD(Altium Designer)软件完成了硬件电路的PCB电路绘制，并且完成了实物的焊接制作，由于软件程序尚未完成，所以只能针对每个模块进行相应的测试，在保证模块正常工作的情况下，进行接下来的软件部分的设计。

# 第4章 软件设计

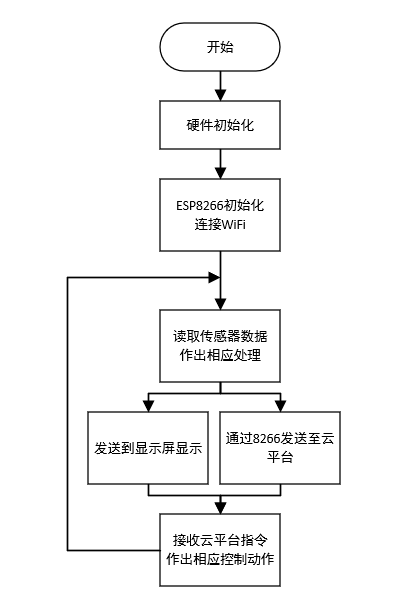
## 软件设计分析

本章将在硬件电路设计、绘制并调试成功的基础上，对软件部分作出具体的分析、规划和编写。整个智能家居模拟系统的软件部分，分为硬件装置的应用程序和网站终端展示页面的程序。

### 硬件装置应用程序结构分析

想要实现设计要求之中的功能，首先应该对所有硬件完成初始化，其中值得一提的是对WiFi模块ESP8266的初始化，在这个模块的初始化应该完成对本地无线路由器WiFi连接，这样才能在正式的循环体中的通过ESP8266发送数据通入互联网。其次就是完成对各传感器的输出数据收集和处理，然后才是将数据发送至显示屏显示和通过ESP8266传入云平台。这样就完成了数据的单向传输。

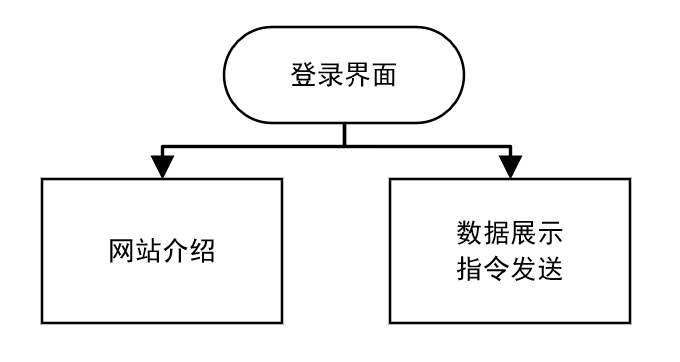
还应该在程序的某个合理的位置，设置对云平台指令的接受、解析和执行。由此，完成数据的双向通信。硬件应用程序流程图如图4-1所示：



**图4-1 硬件应用程序流程图**

### 网站终端展示程序结构分析

网站的实际用途是向用户展示远程连接的传感器数据以体现家里的真实情况，并通过网站发送一些指令来达到简单的远程控制效果。所以首先，这是一个倾向私人化的网站程序，需要创建一个登录界面验证网站使用者的合法性。其次是就是一个展示数据页面，展示数据页面会包含发送指令的功能。最后，因为这是一个毕业设计作品，可以创建一个界面用作网站的简单介绍。



**图4-2 网站展示程序框图**

## 硬件应用程序设计

### 初始化程序设计

此次设计中用到的元器件很多，并且根据各个元器件的功能特性，必须在初始化时设置固定的先后顺序，才能在全部初始化完成后，顺利使硬件装置正常工作。

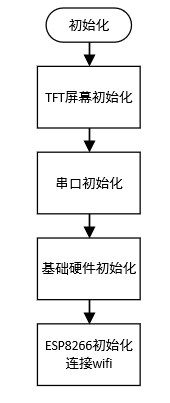
首先是屏幕的初始化，因为在初始化过程中，需要使用屏幕打印出每一步的具体进展，避免用户在硬件初始化期间无法判断装置是否正常工作，以及WIFI是否正常连接。

接着是STM32核心板上的串口1和串口2的初始化，串口1用作与WIFI模块ESP8266的通信，串口2用作打印关键步骤信息，方便调试。

然后便是STM32核心板上led灯、ADC通道初始化、RGB灯、电机以及各传感器等等的初始化设置，这些硬件的初始化很大一部分是要初始化对应的IO引脚。

最后便是WIFI的设置与初始化。ESP8266模块将在此进行硬件初始化，并进行预设的WIFI连接。

初始化部分程序流程图如图4-3所示：



**图4-3 硬件初始化程序流程图**

下面是使用8266官方发布的AT指令集进行WIFI连接的代码，(其中printf通过串口2打印至电脑串口助手帮助调试)：

void WIFI\_CWJAP(void)

{

printf("AT\r\n");

while(ESP8266\_SendCmd("AT\r\n\r", "OK", 200))

delay\_ms(500);

printf("CWMODE\r\n");

while(ESP8266\_SendCmd("AT+CWMODE=1\r\n", "OK", 200))

delay\_ms(500);

printf("AT+CWDHCP\r\n");

while(ESP8266\_SendCmd("AT+CWDHCP=1,1\r\n", "OK", 200))

delay\_ms(500);

printf("CWJAP\r\n");

while(ESP8266\_SendCmd(ESP8266\_WIFI\_INFO, "GOT IP", 200))

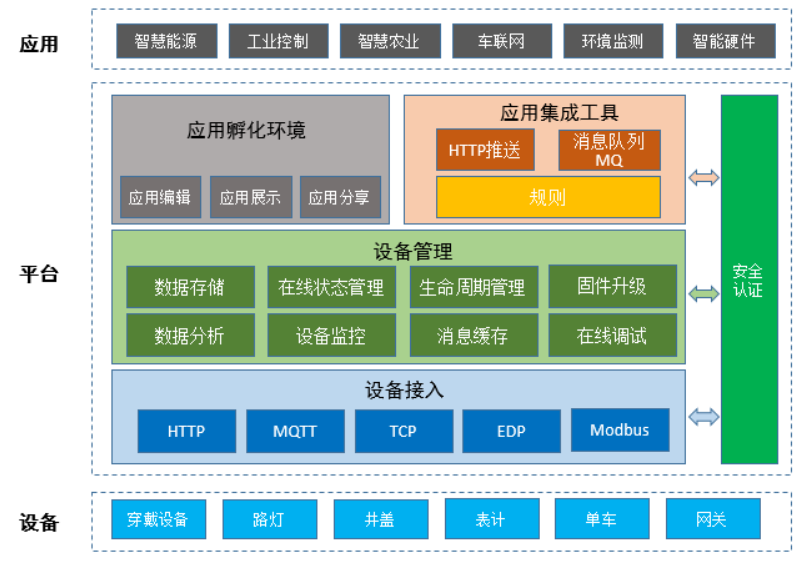
delay\_ms(500);

}

紧接着ESP8266初始化的就是与OneNET平台连接，有关OneNET平台的连接将在下面几节进行详细的说明。

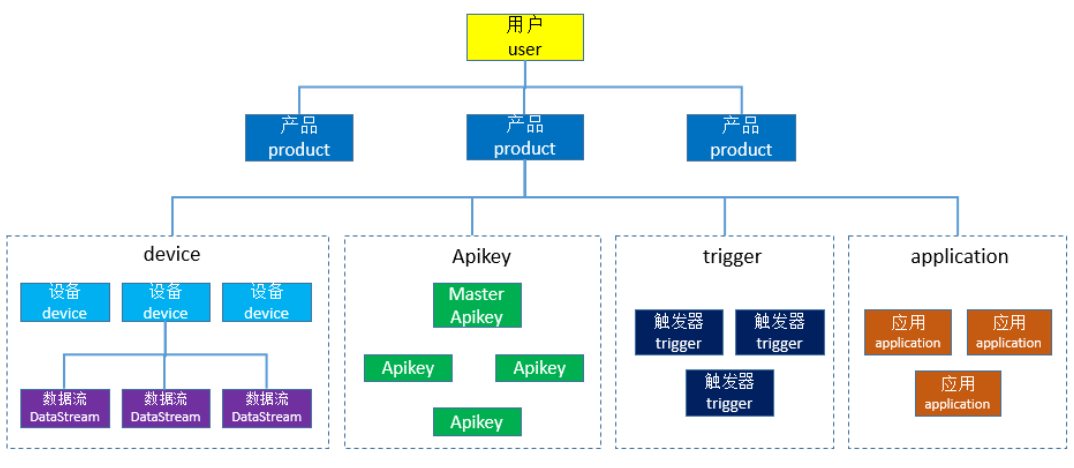
### OneNET云平台简介

OneNET云平台提供包括MQTT、Modbus、HTTP、TCP等等多种协议标准接入，也就是表明OneNET的服务器不会在意你具体用的什么协议，重要的是使用该协议规范正确设置发送和接受的数据包格式。只要成功接收到数据，云平台就能使用其自身数据储存、数据分析能力，在开发者提前设置好的情况下实现数据对接。云平台的功能十分强大，成功数据对接之后还能直接在云平台上进行应用编辑和展示，即对你的设备上传的数据以应用的形式作图形化的展示，非常简单和便捷，但是不能实现高度自定义的界面。

云平台具体的产品接入架构图如图4-4所示：

**图4-4 产品接入架构图**

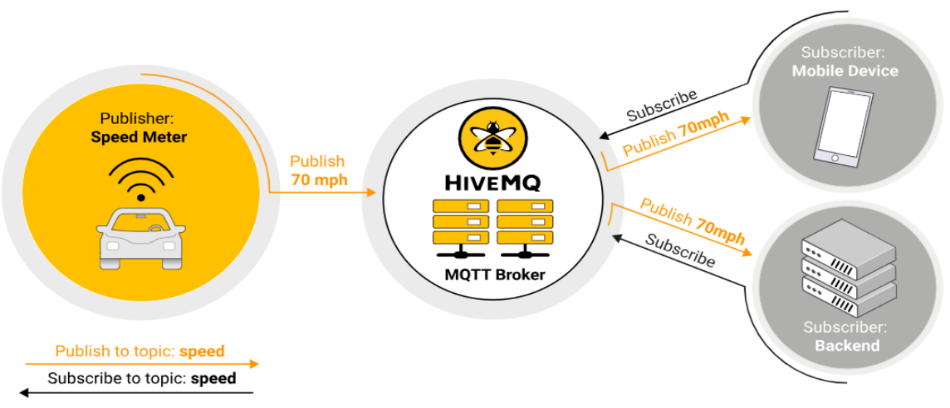
在OneNET云平台上，每个注册的个人用户可以拥有最多十个产品，产品是OneNET上最大的概念，每个产品可以创建很多个设备，而设备则是对应每一个向平台发送数据的硬件装置，每个设备可以创建很多的数据流，因为OneNET向每一个用户暴露的API都是同一个基地址——“183.230.40.39”，所以在创建产品、设备时都会生成一个唯一的APIkey,在每个HTTp请求后面都会附带相应的APIkey，确保正确访问。

云平台资源模型图如图4-5所示：

**图4-5 云平台资源模型图**

### MQTT协议简介

MQTT是一个轻量级的、开放的数据传输协议，因为轻量，所以常用M2M(Machine to Machine)和物联网(IOT)。它并不是制定数据要如何传输，而是制定了数据在传输过程中的数据包格式，和制定了传输模式是发布/订阅消息的模式。

发布/订阅模式与常见的HTTP传输的模式相比，前者并不需要保证客户端必须先于服务器验证连接是可靠和有效的，并且不用一直保持连接，而是通过一个代理作中间件，将客户端与服务器从时间和空间上解耦，客户端和服务器端不需要了解彼此，不需要直接连接。如图4-6所示：

**图4-6 MQTT协议发布/订阅模式图**

### MQTT协议数据包格式

根据MQTT协议官方文档，每个MQTT数据包都应该包含三个部分，即：

**表4-1 MQTT数据包组成表**

|  |  |
| --- | --- |
| Fixed Header | 所有packet中都必须有 |
| Varable Header | 部分包含有 |
| Payload | 部分包含有 |

1. 首先是Fixed Header

**表4-2 MQTT数据包-固定头格式表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

MQTT Packet Type指的是你所使用的MQTT协议版本所支持的数据包类型

以OneNET使用的3.1.1为例，共有14种类型：

**表4-3 MQTT数据包类型**

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 值 | 流向 | 描述 |
| CONNECT | 1 | C->S | 客户端请求与服务端建立连接 |
| CONNACK | 2 | S->C | 服务端确认连接建立 |
| PUBLISH | 3 | C⬄S | 发布消息 |
| PUBACK | 4 | C⬄S | 收到发布消息确认 |
| PUBREC | 5 | C⬄S | 发布消息收到 |
| PUBREL | 6 | C⬄S | 发布消息释放 |
| PUBCOMP | 7 | C⬄S | 发布消息完成 |
| SUBSCRIBE | 8 | C->S | 订阅请求 |
| SUBACK | 9 | S->C | 订阅确认 |
| UNSUBSCRIBE | 10 | C->S | 取消订阅 |
| UNSUBACK | 11 | S->C | 取消订阅确认 |
| PING | 12 | C->S | 客户端发送PING(连接保活)命令 |
| PINGRSP | 13 | S->C | PING命令回复 |
| DISCONNECT | 14 | C->S | 断开连接 |

1. Varable Header ，由于是部分包含有，所以此处以第一种CONNECT类型举例

**表4-4 CONNECT类型灵活头格式**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1-2 | ProtocolName Length | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| byte 3 | ‘M’ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| byte 4 | ‘Q’ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| byte 5 | ‘T’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| byte 6 | ‘T’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Byte7 | Protocol Level | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Byte8 | Connect Flag | User  flag | Password  flag | WillRetain  Flag | WillQos Flag | | Will  Flag | CleanSession  Flag | Reserve |
| Byte9-10 | KeepAlive |  |  |  |  |  |  |  |  |

（1） user flag与password flag

平台不允许匿名登陆，因此这两个标志位在连接时必须设置为1，否则认为协议错误，平台将会断开连接。

（2） Will flag与Willretainflag/WillQosFlag

Will flag 表示客户端如果不是在发送DISCONNECT中断的通信，在设置了WillRetain Flag 和WillQos Flag 时，执行重传操作

（3） CleanSessionFlag

若客户端将clean session标志位设置为0，当其断开后，平台将会保存session，反之不会保存

（4） Reserve 保留位，置0。

（5） KeepAlive保活时间

每个客户端可自定义设置连接保持时间，最短120秒，最长65535秒。

1. Payload

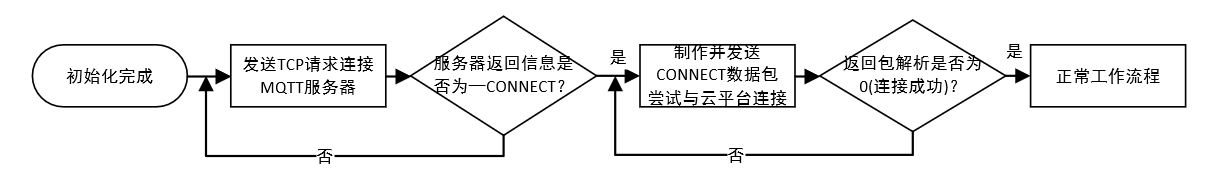
**表4-5 CONNECT类型载荷格式**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Description** | **是否必须存在** | **格式** |
| Field1 | Client Identifier | 是 | 2字节字串长度 + utf8字串 |
| Field2 | UserName | 是 | 2字节字串长度 + utf8字串 |
| Field3 | UserPassword | 是 | 2字节字串长度 + utf8字串 |

在OneNET上 Client Identifier对应平台创建设备时的设备id，UserName对应产品ID，password为对应的鉴权信息（auth\_info）(这是一个创建产品时由创建者设置的特殊字段)

综上，为MQTT数据包格式简单解析。

### 云平台通信程序设计

第一个小节中，完成ESP8266的初始化之后，按照云平台的文档说明，需要先于云平台制定的MQTT服务器进行TCP连接。在得到服务器反馈表示连接成功之后，才能在MQTT协议框架下尝试发送CONNECT类型的数据包与OneNET平台进行连接。同样，在得到云平台的MQTT数据包并解析成功，内容表示连接成功，才能正常的发送数据和接受指令。云平台通信程序流程图如图4-6所示：

**图4-7 通信程序流程图**

### 云平台上传与接收程序设计

将数据上传至云平台的大致流程与前一小节有些类似，只不过不需要在已经连接成功的基础上再连接一次。而是先根据MQTT协议的发布数据(PUBLISH)类型,制作相应数据包，将数据填充至相应位置后，再通过8266发送。至于为什么不直接在制作数据包的过程中直接写入数据，是因为在以这种思路尝试时出现了数据发送不完整的情况，所以将制作数据包封装成独立函数，并解耦只在数据包制作成功后再填充数据。

在主程序main函数中每个while循环的开始，都会先通过8266尝试获取云平台发回的数据，如果接收函数返回不为空，则会进入MQTT包解析函数解析并提取数据，根据设定的阈值作出相应的改变实现控制指令。

## 网站程序设计

### 搭建服务器环境

一般在本地制作的网站都只能在本机上运行，没有服务器环境和相应的配置无法实现外网访问我们挂载的网站。个人搭建服务器环境通常有两种解决方案：

1. 在本机环境下创建虚拟机，并将虚拟机配置为服务器，使用花生壳等工作做内网穿透，如果本机网络是由路由器管理，那还需要再路由器上设置相应的映射规则。
2. 通过云服务商如阿里云、腾讯云、七牛云等购买云服务器服务，购买的云服务器会被云服务商分配一个固定的外网可访问的IP地址，使用此IP地址就能进行 网站环境的搭建。

本设计中将两个方案都一一尝试过一遍，在第一个方案中实现了WEB服务器配置和无线路由器作为网关的局域网内的虚拟机服务器访问，在尝试做规则映射时失败，原因未知，且网上没有找到对应的解决方法，只能放弃。在尝试第二个方案时，发现阿里云对学生身份的开发者，可以在完成阿里云官方设计的云服务器轻度学习任务之后领取免费三个月的2核4GECS云服务器。

在服务器环境能正常使用的情况，直接将网页源码挂载到服务器上是不能生效的，即在浏览器中通过IP访问还是会失败，因为此时的服务器是一个纯粹的带操作系统的机器，并没有进行WEB服务环境的配置。

本设计中使用APACHE作为基础WEB环境，在CentOS6.5环境下搭建Apache httpd服务器。一共分为简单的三步。

1. 在CentOS下安装Apache httpd，命令行输入yum -y install httpd\*，其中-y这个参数是指安装过程提示选择Yes or No时，全部自动选择为Yes：\*号表示httpd下的所有文件
2. 在CentOS根目录下找到var文件夹中的www文件夹(或许有些在www里面还有个html文件夹)，将准备好的网页源码复制进去
3. 启动httpd服务器，命令行输入service httpd start ，再去阿里云的控制台中找到安全组选项，设置放行常用的80、8080、443、22端口号。设置好通过浏览器访问IP地址就能成功看到网站的正确显示。

### VUE框架简介

VUE是一套用于搭建用户界面的渐进式框架，即用到什么部分就加载什么部分，不会造成工程的臃肿和不必要的编译耗时。VUE这类前端框架可以用常见的MVVM框架(Model-ViewModel-View)模型去理解，且与其他常见框架不同的是，VUE只关心视图层，将注意力集中保持在核心库，而将其他功能如路由和全局状态管理交给其他相关的库。

VUE内部使用VM(Virtual DOM)作为核心处理对象，将逻辑层的操作结果放入VM,再将VM通过指定挂载DOM节点渲染到页面上。在日常开发上提供响应式（Reactive）和组件化（Composable）的视图组件，还有符合ECMAScript规范的一些常用指令，如v-if、v-show、v-on、v-bind、v-for等等，为前端开发带了不少便利。

### VUE-CLI创建VUE工程

为了使各方面趋于网页设计的规范，并且能方便使用VUE的工具库进行开发，本设计中决定使用VUE官方推出的脚手架工具vue-cli进行前端工程的快速开发。

Vue-cli是基于webpack 和 webpack-dev-server 的创建出的脚手架工具，开发者可以使用此脚手架在命令行界面使用特定命令快速构建一个基于个人自定义配置的vue工程模板。

在下载vue-cli之前，需要先下载nodejs并安装才能使用npm 安装。按VUE官方的提示，nodejs应安装v8.9及以上的版本才能实现对vue-cli的支持，这里我选择的是nodejs中文官网上有完整中文文档的v14.16.1版本。

在创建好的目录下进入命令行，输入命令 vue create hello-world ，cli创建工程一般有两种选项：1.default(babel,eslint) 2.manually select features 第一种就是默认选项，以最小化思想只装入必备的babel(将ES6语法转成ES5)和eslint(语法检查)；第二种是自定义要安装的模块。这里选择第二种，因为此次网站设计需要在不同页面之间切换，需要用到vue-router，而这也恰好符合VUE渐进式框架的特点。

正确选择并创建工程成功之后，使用VSCODE打开就能看到一个基础的前端工程框架。工程目录结构如下所示

├── dist // 打包输出文件夹

├── src // 源代码

│   ├── assets // 主题 字体等静态资源

│   ├── components // 全局公用组件

│   ├── router // 路由

│   ├── pages // 主要页面代码

│   ├── App.vue // 入口页面

│   ├── main.js // 入口 加载组件 初始化等

├── .babelrc // babel-loader 配置

├── eslintrc.js // eslint 配置项

├── .gitignore // git 忽略项

├── public //放置vue的虚拟dom挂载的html

│ ├── favicon.ico // favicon图标

│ ├── index.html // html模板

└── package.json // package.json

### 网站布局设计

在制作一个网站的过程中，设计页面布局是一个非常困难的部分，因此本设计的网站布局设计选择参照GitHub上的开源设计STISLA的页面整体布局设计和网站整体颜色风格。

网站整体分为顶部导航栏和主要内容区，在两者之间加入了面包屑路径导航栏，而每个页面的切换则会体现在主要内容区和面包屑上。

## 本章总结

本章通过对硬件程序的设计和分析，梳理出了硬件应用程序的大致流程，并针对其中比较关键和比较困难的部分——硬件初始化、云平台的通信程序作出了比较详细的分解和阐述。MQTT协议是云平台支持接入的多协议中的一种，所以在正式阐述云平台通信程序之前必须对协议的数据通信模式和协议数据包格式作简单介绍，才能更好理解本地接入云平台流程。

# 第5章 结果分析

## 设计指标测试

通过控制变量法，在空气流通良好的学生寝室内和自习室内做了表5-1和表5-2两个实验结果,根据通风情况的不同在学生寝室和自习室内选取了三个位置进行测试，每个位置都是读取的静置5分钟之后的读数，不难看出，在装置正常工作时，读取到的温度和湿度示数与温湿度计示数误差相差都在指标范围内，满足了设计要求。

**表5-1 实验结果1**

|  |  |  |  |
| --- | --- | --- | --- |
| 条件：学生寝室 | | | |
| 设计指标：温度±1°，湿度±2% | | | |
| 位置/通风情况 | 饮水机前(通风良好) | 同学A书桌上(通风较差) | 同学B书桌上(通风一般) |
| 温湿度计 | 25.5°/ 65% | 25.0°/65% | 25.0°/65% |
| 硬件装置 | 25.4°/ 66.1% | 25.8°/64.4% | 25.4°/66.4% |

**表5-2 实验结果2**

|  |  |  |
| --- | --- | --- |
| 位置/通风情况 | 窗前(通风良好) | 死角(通风较差) |
| 温湿度计 | 25.1°/ 63% | 25.2°/64% |
| 硬件装置 | 25.4°/ 64.6% | 25.7°/65.2% |

由于自习室内窗户都是开着的，所以只取了死角和窗边两个点

## 调试过程中遇到的问题

1. 因为在执行循环体内的程序时有很多需要延时等待的地方，并且获取平台下发的数据这个动作在循环体的开头，所以执行推送数据到显示屏和云平台这个动作需要设置一个时间间隔，去尽可能的缩短每次执行循环体所用的时间。如果执行间隔越短，获取控制命令的成功率越低，间隔越长，成功率越高，但是显示屏和云平台数据刷新的频率也越慢，在一定程度上会影响使用者的实际体验。

所以需要作出多组时间间隔比较，选出一个中肯的数值。

1. 在布局、绘制PCB板时没有考虑到一个问题：因为烟雾传感器MQ系列，在正常工作时是需要一个加热电流进行加热的，所以如果MQ系列传感器距离温湿度传感器太近，当MQ系列传感器发热时，读取到的温度示数会有误差，且误差会让温度读数超出指标范围，尽管湿度的读取与温度有关，但是实际表现湿度偏差并不大。根据多次的测量装置正常工作前一分钟的温度均值与正常工作5分钟之后的温度数值比较，得出温差大约在1.1°左右。解决方法自然是在显示和发送之前，对数据作温度补偿。

# 第6章 总结与展望

## 本文总结

为完成本次设计的所有功能，需要对STM32单片机、TCP协议、MQTT协议以及网站制作和服务器相关知识有比较深刻的认识。并且在开始实际制作之前，需要对设计整体根据指标和具体要求作出比较细致的规划，通过网上材料、文献以及咨询导师确定各部分方案的可行性，只有大方向正确，在细节处下的功夫才不会白费。

毕业设计不同于以往的做过的课程设计，不仅工作量大大超出以往的课程设计，大部分内容与我而言都是很新的知识，如STM32、MQTT协议和网站制作使用的JavaScript语言和VUE框架等等。只有合理的规划和安排才能在有限的时间内作出比较高质量的作品。本次设计中，在硬件方面我花了很多时间去选择使用的传感器型号，并使用Altium Designer进行器件原理图与PCB板的绘制，由于使用经验确实不多，在很多细节的地方花了许多不必要的时间，尽管最后板子能正常工作，但还是出现了焊盘比实物小，无法焊接；元件不好的布局导致传感器检测有固定误差等问题，能补救是令人庆幸的，但是也要长记性，记住在这些方面的教训。软件方面经过前期的STM32芯片的学习，能顺利使用芯片完成基础功能的实现，也为正式编写硬件程序打下基础，后期经过多方文档的样例程序和不断的调试、修改实现了设计要求中的功能。另一方面，在经过了对网站制作的有关知识JS、HTML、CSS以及框架的学习大致了解了制作方法，并且在制作demo网站之后认识到必须要加上服务器环境的相关配置才能实现网站的互联网访问，继而学习了简单的服务器相关知识。

总的来说，在实现了硬件和远程网页的联调之后，更加深刻的认识到智能家居对于生活所带来的的便利，和更广阔的扩展空间，自己的设计本身也还有许多不足需要改善和加强。

## 工作展望

在设计之初。进行方案规划时，对于入户监测功能的实现希望用摄像头加上人工智能中比较经典的人脸识别实现，但是在可行性上因为涉及到了另一个领域专业知识，变得很低，希望能在今后不仅能继续完善智能家居系统的功能实现，更可以进行多学科、多领域的大胆尝试，将诸如人工智能AI、现今火热的5G以及云计算等融合并实现在智能家居上的应用。

在查看文献时将自己的方案与别人的方案作对比，就能清晰地意识到自己只是实现了真正意义上智能家居系统的本地网络中的一个功能比较综合的节点，实际上既然是智能家居那么必然要做到对于整个屋内的环境监测，相应的，多节点的组网模式也是必须的，才能针对各节点的例如温度数据作平均或加权平均最终得出一个比较接近的室内温度，而这也是今后需要努力的方向之一。

功耗问题其实也是一个物联网设计必须要加以考虑和实现的方面，从哪一方面可以切入功耗问题，如何实现低功耗都是值得深入研究的问题，也是未来本设计需要完善的地方。

最后，其实本次设计本身并没有针对真正的家电设备作出相应的控制动作，则对于如何让各个厂商不同家电不同规格的都能接入智能家居系统，也是一个十分重要的工作。

# 

# 致谢

该论文是在于春梅老师的指导下完成的。我对于智能家居这个题目也是在老师将课题给我之后，第一次进行了比较全面的了解和认识。因为我对智能家居的理解是逐渐加深的，所以每次思考该怎么设计整体方案和布局都会有新的认识和新的考量，会有无数的想法在我的脑子里盘旋，我认为这是一个非常灵活的题目，也是一个深刻而充满趣味的领域，因为它是在众多学科不断高涨的浪潮下诞生的。在和于老师的多次沟通之下，结合我自己擅长的领域，才终于是确定了大致的制作方向。对于我们在毕业设计期间的所有问题，于老师和身边的同学们都会尽力给予我们帮助，鼓励和肯定。在我每次都怀疑自己的时候，于老师都会让我们继续坚持下去，认真做好规划和分析，争取走好每一步。除此之外，于老师也会每周定期和我们交流进度和遇到的困难，督促我们在看似漫长实则转瞬即逝的这段时间里，不浪费一分一秒。

从一开始和老师见面，老师就在感慨时间会过得很快的，到如今，我的毕业设计也要进入下一个阶段了。回顾此次毕业设计和大学四年的学习生活，我经历过很多课程设计和比赛，有难有易，但从来都没有一蹴而就的事情，只有认真做好准备、深入地去探索和研究，才能在真正开始时游刃有余。我在这四年中最能学以致用的可能就是在做之前仔细地根据现有的资料和自己的经验去分析解决的思路和方案，拿出可行的，作出在情景下的最优解，而不是一味地盲目随波逐流。

感谢一路以来给予我无私帮助的每个人，谢谢和蔼的老师们，谢谢热情的同学们，我相信这段经历一定会受用终生，让我在今后更好地去面对挑战与未知。

此致，敬礼。

# 参考文献

1. 刘辰帅. 智能家居行业存在的问题与建议[J]. 城市建设理论研究:电子版, 2015, 000(030):296-297.
2. 陈思运,刘烃,沈超,苏曼,高峰,徐占伯,师嘉悦,贾战培.基于可穿戴设备感知的智能家居能源优化[J].计算机研究与发展,2016,53(03):704-715.
3. 马跃其. 基于ZigBee无线通信技术的智能家居系统[D].河南理工大学,2010.
4. F Akyildiz, W Su, Y SanKarasubramaniam, et al. Wireless Sensor Network[J]．Computer Networks，2002，38(4)：393-422.
5. Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Cullar, John Anderson. Wireless Sensor Networks for Habitat Monitoring．2002 ACM International Workshop on Wireless Sensor Networks and Applications，September 2002.
6. 龚江涛,陈金鹰,方根平. Zigbee技术特点及其应用[A]. 四川省通信学会.四川省通信学会2005年学术年会论文集[C].四川省通信学会:中国通信学会,2005:4.
7. 郦亮. IEEE 802.15.4标准及其应用[J]．电子设计应用，2003：18～26.
8. 张茁,孙洁.基于以太网的智能家庭网络系统设计[J].计算机工程与设计,2005(11):275-276+303.
9. 赵亮. 智能家居语音控制系统的设计[D].西安电子科技大学,2017.
10. 喻金,肖金凤,宋玲玲,于红利,廉莉莉.ZigBee技术在智能箱式变电站中的应用[J].物联网技术,2012,2(11):32-35.
11. 感烟火灾探测器迟滞时间模型的理论和实验研究[D]. 中国科学技术大学, 2005.
12. 刘恺乐. 基于单片机的火灾报警器设计[J]. 数码世界, 2016, 000(010):92-93.
13. 林钰恒. 基于GMR传感器和无线传感网的车位检测系统设计[D]. 杭州电子科技大学.
14. Mendez G R, Yunus M A M, Mukhopadhyay S C. A WiFi based smart wireless sensor network for an agricultural environment[C].International Conference on Sensing Technology. IEEE, 2012:405-410.
15. 徐文. 基于WiFi与Android的智能家居监控系统设计[D]. 西南交通大学, 2017.
16. 蒋开伟, 孙凌杰. 基于云平台的智能家居控制系统设计[J]. 电子世界, 2018, No.555(21):141-142.
17. 晓风. 火灾探测技术及发展趋势[J]. 安防科技, 2003, 000(002):18-20.
18. 蔡健. 大型临床医疗设备智能管理平台的设计与实现[D]. 东南大学, 2018.
19. 王秋颖. 基于超声波的汽车倒车雷达系统研究[D]. 天津大学.
20. 汪建文. 一体化校表机器人末端执行系统的研究及其实现[D]. 重庆大学.
21. 阮进军, 孙握瑜, 朱先远. 一种基于LoRa窄带物联网技术的鱼塘监控系统设计与实现[J]. 齐齐哈尔大学学报:自然科学版, 2019(6):8-11.
22. 邓成, 孙书会. MVVM设计模式的前端应用[J]. 电脑知识与技术, 2019(29).